Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Autonomous Systems 54 (2006) 23–33
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

A higher level path tracking controller for a four-wheel differentially steered mobile robot

Elie Maalouf[a], Maarouf Saad[a,*], Hamadou Saliah[b]

[a] *Electrical Engineering Department, Université du Québec, École de technologie supérieure, Montréal, Québec, Canada*
[b] *Télé-université, Université du Québec, Montréal, Québec, Canada*

Received 2 July 2004; received in revised form 4 October 2005; accepted 6 October 2005
Available online 23 November 2005

Abstract

A variety of approaches for path tracking control of wheeled mobile robots have been implemented. While most of these are based on controlling the robot dynamics, they are not applicable if the robot dynamics are inaccessible. In this paper, a fuzzy logic controller (FLC) for the path tracking of a wheeled mobile robot based on controlling the robot at a higher level is presented. The controller is highly robust and flexible and automatically follows a sequence of discrete waypoints, and no interpolation of the waypoints is needed to generate a continuous reference trajectory. The speeds are varied depending on the variations in the path and on the posture of the robot. The heuristic rules of the FLC are based on an analogy with a human driving a car and the optimization of the controller is based on experimentation. The implementation on a P3-AT mobile robot shows the effectiveness of the proposed approach.

Keywords: Path tracking; Fuzzy logic; Fuzzy control; Fuzzy control language; Look-ahead curvature

1. Introduction

In mobile robot navigation, the path tracking controller is usually implemented at a low level of the control hierarchy. Its function is to execute a path planned by a higher level path planner with the least possible error in position and with minimal control effort. The function of the high level planner is to plan a path either offline or online depending on changes in the environment. The path generated could be similar to following some virtual robot. The lower level path tracking controller guarantees that the robot will follow the path in a precise, reliable, and efficient manner. The path following problem is highly nonlinear, and several approaches have been developed to solve the problem of path tracking through direct control of the robot's dynamics. In some of these approaches [1, 2], nonlinear controllers are derived based on the Lyapunov approach. Other types of controllers were designed [3–7] using sliding mode control or other nonlinear techniques and were applied to the nonlinear dynamic and kinematical model of the robot. A behavioral approach [8] for path tracking was implemented using fuzzy logic control and was implemented for wheel steering.

Irrespective of the performance of these approaches, they cannot be implemented if the robot dynamics are inaccessible. If no direct control on motor torques and traction forces can be done, such techniques cannot be used. A controller at a higher level can be used to solve this problem. Motion is controlled using the kinematical model of the robot as the system. The control law has to respect the kinematical constraints. The variables calculated by the control law are the translational and rotational velocities, based on the position, orientation, and the current values of the translational and rotational velocities. Here is a brief overview of some of the control techniques for control at the higher level on the kinematical model.

In [9], a controller is implemented using a biologically inspired shunting model, which is integrated into a bang–bang controller. In another approach [10], a controller is designed using a back-stepping technique. In [11] the stability of a pure pursuit path tracking algorithm is analyzed for a kinematical model using a linearized kinematical model. The analysis is done for the case of a straight line and a circle, with the

* Corresponding author. Tel.: +1 514 396 8940; fax: +1 514 396 8684.
E-mail address: maarouf.saad@etsmtl.ca (M. Saad).

0921-8890/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2005.10.001

reasoning that most trajectories can be decomposed into pieces of constant curvature.

A generalization of the quadrature curve approach [12] has been implemented. The idea is to make the robot follow a quadratic curve to a reference point on a desired path. The reference point is moved in time until the goal destination is reached. A path tracking algorithm that uses a scalar controller [13] based on static path geometry with position feedback has been implemented on three types of wheeled mobile robots, one of which is differentially steered.

Despite the interesting features of all these controllers, they are difficult to tune, in contrast to the flexibility that fuzzy logic control provides. Fuzzy logic control (FLC) is an interesting tool to be applied to the problem of path tracking since the output varies smoothly as the input changes. In this article, we will discuss a fuzzy path tracking controller designed based on expert experience and knowledge that was implemented on a four-wheel differentially steered mobile robot. The fuzzy inference rules are based on reasoning similar to that of a human driver on a road that is free of obstacles and other cars. If the road is straight, the driver can displace at higher speeds. When faced with a curvature, he gradually decreases speed and makes a smooth turn. The behavior of the human driver is apparent in the fuzzy inference rules. The membership functions were derived based on the kinematical constraints of the robot. The built-in PID controllers were used to control the vehicle dynamics. Each wheel is controlled separately, and a user can only change the gains of the PID, which is virtually futile for the purpose of path tracking.

The fuzzy controller takes in the desired path in the form of a list of discrete waypoints and drives the robot in proximity of those waypoints. This has the benefit that no continuous trajectory needs to be calculated using an interpolation technique, which would be very advantageous if the desired path changes in real time. If a precise path needs to be followed, the distance between the consecutive waypoints of the trajectory can be reduced so as to have better precision. Attainment of higher precision is done at the expense of driving at lower speeds. Another advantage of the fuzzy controller implemented in this paper compared to other techniques for trajectory following is that control is done only to follow reference waypoints and no reference velocities and heading are imposed. This renders the fuzzy path tracker more robust since control is done on only one parameter. The fuzzy controller is intelligent in the sense that it can determine the control outputs without needing a continuous reference trajectory or reference speeds and heading. Moreover, the fuzzy controller applied in this article can be modified for navigation on rough terrain.

If the built-in controller at the dynamic level doesn't perform properly in accordance with the controller at the kinematical level due to variations in the physical properties of the surface, an adjustment technique can be used to sidestep the problem. An intelligent predictive control approach that adapts the reference inputs (velocities) based on real-time learning [14] is one good technique to be explored and added in cascade after the path tracking controller in the control loop if the robot dynamics are inaccessible.

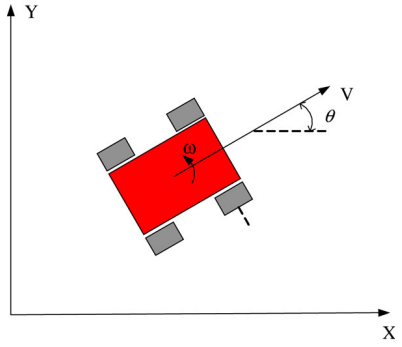

Fig. 1. Robot kinematical parameters.

The organization of the remaining part of the paper is as follows. Section 2 is a concise description of the kinematical modeling of differentially steered wheeled mobile robots. In Section 3, the path tracking parameters of the controller are described as well as some notions that were used to design the controller. In Section 4, the different modules of the path tracking controller are presented. In Section 5, a classical controller that was implemented for a comparison with the fuzzy controller of this article is presented. In Section 6 the implementation of the controller as well as the robot interface that was used are described briefly. In Section 7, simulation and real time experimental results of the fuzzy and classical controller are presented for several sequences of waypoints, along with a sensitivity analysis for possible discontinuities in a trajectory.

2. Kinematical model

The modeling of the kinematics of differentially steered wheeled mobile robots in a two-dimensional plane can be done in one of two ways: either by Cartesian or polar coordinates. The modeling in Cartesian coordinates is the most commonplace and the discussion will be limited to modeling in Cartesian coordinates. The robot has four wheels and is differentially driven by skid steer motion.

The motors that power the wheels at each side are geared internally to ensure that the velocity of the two adjacent wheels at each side are synchronized (having the same angular velocity) and thus have the same velocity at ground contact.

The posture of the robot in the 2-D plane at any instant is defined by the position in Cartesian coordinates and the heading with respect to a global frame of reference. The kinematical model is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V\cos\theta \\ V\sin\theta \\ \omega \end{bmatrix} \tag{1}$$

where x and y are the coordinates of the center of the robot, θ the heading and is positive counter-clockwise, V and ω the linear and angular velocities respectively (Fig. 1).

3. Path tracker parameters

The path tracking controller that was implemented here is based on the controller in [15] but with some major changes in

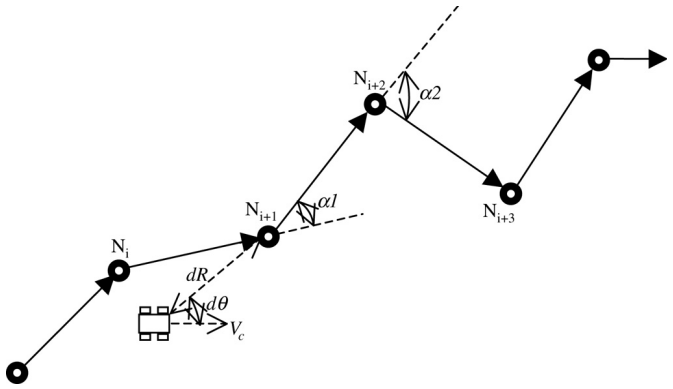

Fig. 2. FLC parameters.

the inputs and outputs of the fuzzy logic controller (FLC) and the rules, as well as in the path representation. The form of the control law equation is as follows:

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} f1(C, \mathrm{d}R, \mathrm{d}\theta, V_c) \\ f2(C, \mathrm{d}R, \mathrm{d}\theta, V_c) \end{bmatrix} \tag{2}$$

where V and ω are the translational and rotational velocities of the robot, C the *look-ahead curvature* (LAC) and is a feed forward input, $\mathrm{d}R$ the distance from the actual position of the robot to the next desired position, $\mathrm{d}\theta$ the difference between the angles of the line joining the current position to the next desired position and the actual heading of the robot, V_c the current linear velocity (see Fig. 2).

The functions $f1$ and $f2$ are the control laws of a Sugeno type fuzzy controller. Sugeno controllers take in fuzzy inputs and discrete outputs. The outputs are calculated separately. First, let's describe the parameters used for the controller. C is obtained using another fuzzy logic module whose inputs are $\alpha1$ and $\alpha2$. In Fig. 2, the input parameters of the controller are illustrated for a particular posture of the robot. The trajectory is described by a set of discrete node positions N_1 to N_{Final} linked to each other starting from the initial position to the final desired position.

The task of the robot is to pass in proximity of these points in the right order in a continuous and smooth manner. A continuous trajectory can be discretized as needed. The behavior of the controller is such that if the discrete points are close to each other, higher precision is achieved but at lower speeds.

If less precision is required, the discrete points can be selected further apart and the robot moves at higher speeds. The current node N_i is defined as the node whose position is nearest to the robot's current position. The next node is N_{i+1} in the list of nodes on the trajectory and N_{i+2} is the one after N_{i+1}. The angles $\alpha1$ and $\alpha2$ are the angles between the lines N_iN_{i+1} and $N_{i+1}N_{i+2}$, and between the lines $N_{i+1}N_{i+2}$ and $N_{i+2}N_{i+3}$, respectively. If $\alpha1$ and $\alpha2$ are large, the robot must slow down to be able to make a smooth turn. C is a parameter that is a function of angles $\alpha1$ and $\alpha2$ used to indicate the steepness of the curvature. The path is represented by a linked list of nodes starting with the start node and ending with the destination node. A pointer to the current node N_i points initially to the first node of the list. Whenever the robot gets nearer to the next node position, N_{i+1}, than the current one, N_i, then N_{i+1} becomes the current node and the robot will always head in the direction of the node N_{i+2}. The pointer would now point to the current node. Consequently, the pointer to the current node can only change incrementally starting from the beginning of the list.

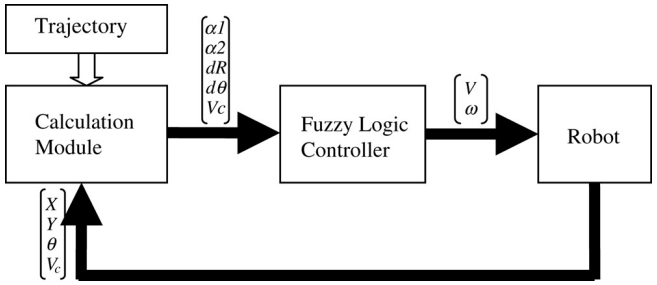

Fig. 3. Control diagram.

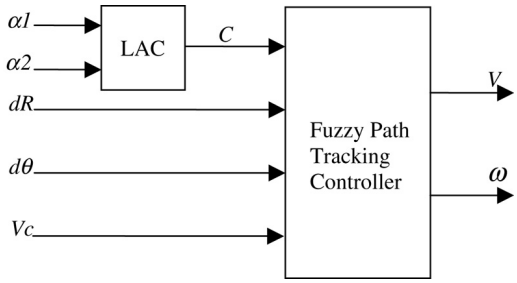

Fig. 4. FLC schematic.

If a robot has a high current velocity V_c, and needs to make a sharp turn $\mathrm{d}\theta$, then it must first slow down while turning smoothly. When it has slowed down sufficiently, the robot can start making a turn in response to the curvature. All the parameters to the input of the controller can be calculated knowing the current node and the robot's current position and heading, as well as its current velocity. The block diagram of Fig. 3 is the general structure of the control loop.

The calculation module takes in the position, heading, and current velocity of the robot, determines the current node state of the robot with respect to the trajectory, and calculates the controller parameters. The fuzzy logic controller then determines V and ω such that the robot follows the trajectory in a smooth and efficient manner.

4. Fuzzy path tracking controller

The task of the path tracking fuzzy controller is to direct the robot to follow the trajectory in a smooth and continuous manner at the best possible precision. It might not be necessary that the robot passes exactly through the points on the trajectory, but at least passes in their proximity and arrives to the final destination. The closer the discrete points are to each other, the more precise the robot will be in executing the trajectory, but at lower speed. In Fig. 4, the schematic of the FLC is shown. The first module determines the numerical value of the look-ahead curvature (LAC) and the path tracker module determines the linear velocity and angular speed, the controller outputs. Both modules are Sugeno type fuzzy inference systems

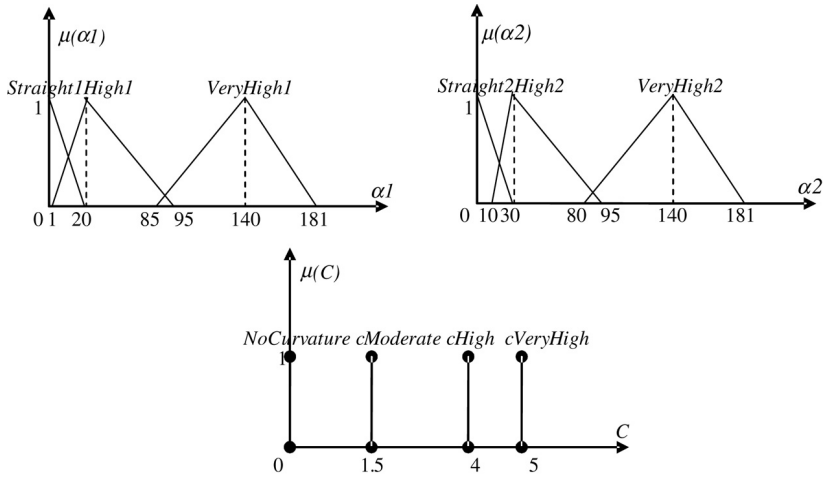

Fig. 5. LAC parameters and their corresponding membership functions.

of order zero. The document [16] contains a brief and practical introduction to fuzzy control. For a more detailed analysis on fuzzy control, [17] provides a more in-depth theoretical study. The LAC uses the angles $\alpha 1$ and $\alpha 2$ to determine the value of C. The membership functions of each of the parameters are shown in Fig. 5. The membership functions of $\alpha 1$ are *Straight*1, *High*1, and *VeryHigh*1 and those of $\alpha 2$ are *Straight*2, *High*2, and *VeryHigh*2. The membership functions of C are singletons that take values between zero and five. The zero value indicates that there is no curvature, meaning that $\alpha 1$ and $\alpha 2$ are small and the robot will follow a straight line at the current state. If the curvature is high, the robot must slow down to make the sharp turn. The inference rules map the membership functions of the input parameters to the membership functions of the output.

For example the rule:

IF $\alpha 1$ is *Straight*1 AND $\alpha 2$ is *VeryHigh*2 THEN C is *cHigh*

maps membership functions *Straight*1 and *VeryHigh*2 of the inputs to membership function *cHigh* of the output. If the two conditions of the inputs are satisfied for this rule, then the value four corresponding to *cHigh* is returned for this rule. The truth value for the rule is obtained by using the product of the truth values of *Straight*1 and *VeryHigh*2:

$$A_i = \mu(Straight1(\alpha 1))\mu(VeryHigh2(\alpha 2)) \tag{3}$$

where μ is a value between zero and one that indicates the truth value that an input value belongs to some membership function. The values of all rules are returned and the output returned by the LAC module is defuzzified using the center of gravity method for singleton (COGS) [16]. The formula of this method is:

$$U(t_k) = \frac{\sum_{i=1}^{p} U_i A_i(t_k)}{\sum_{i=1}^{p} A_i(t_k)} \tag{4}$$

where A_i are the singleton values (i.e. truth values) of the individual rules, and U_i their corresponding outputs. Fig. 6 shows the input–output characteristics of the LAC. It is clear

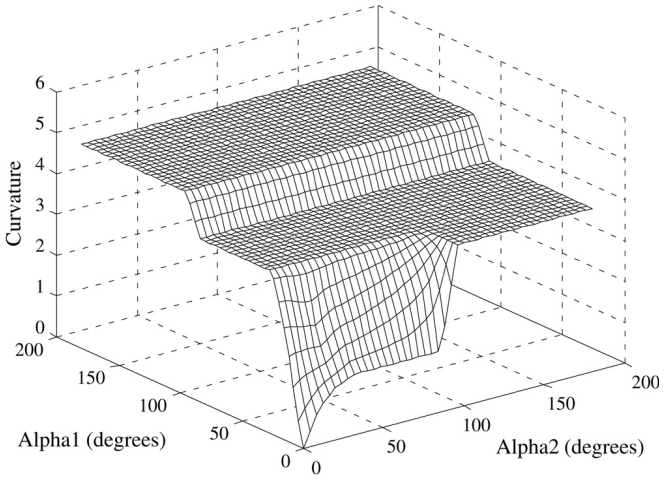

Fig. 6. Input–output surface for LAC.

Table 1
Rule base for the LAC

$\alpha 2$	$\alpha 1$		
	Straight1	High1	VeryHigh1
Straight2	NoCurvature	cHigh	cVeryHigh
High2	cModerate	cHigh	cVeryHigh
VeryHigh2	cHigh	cHigh	cVeryHigh

that the output rises faster as $\alpha 1$ increases. The rule base is shown in Table 1.

The value C is then fed to the path tracking controller along with dR, $d\theta$, and V_c. The membership functions of each of the input and output parameters are shown in Fig. 7. As expected, C ranges from zero to five. Input dR ranges between zero and 7000 mm, $d\theta$ ranges from $-180°$ to $+180°$, and V_c from zero to 1000 mm/s.

The linear velocity output ranges from zero to 800 mm/s and the angular speed from -30 to $+30$ degrees/s. The inference rules maps the input membership functions to the output membership functions. The behavior of the controller is such that it changes linear velocity and angular speed in a

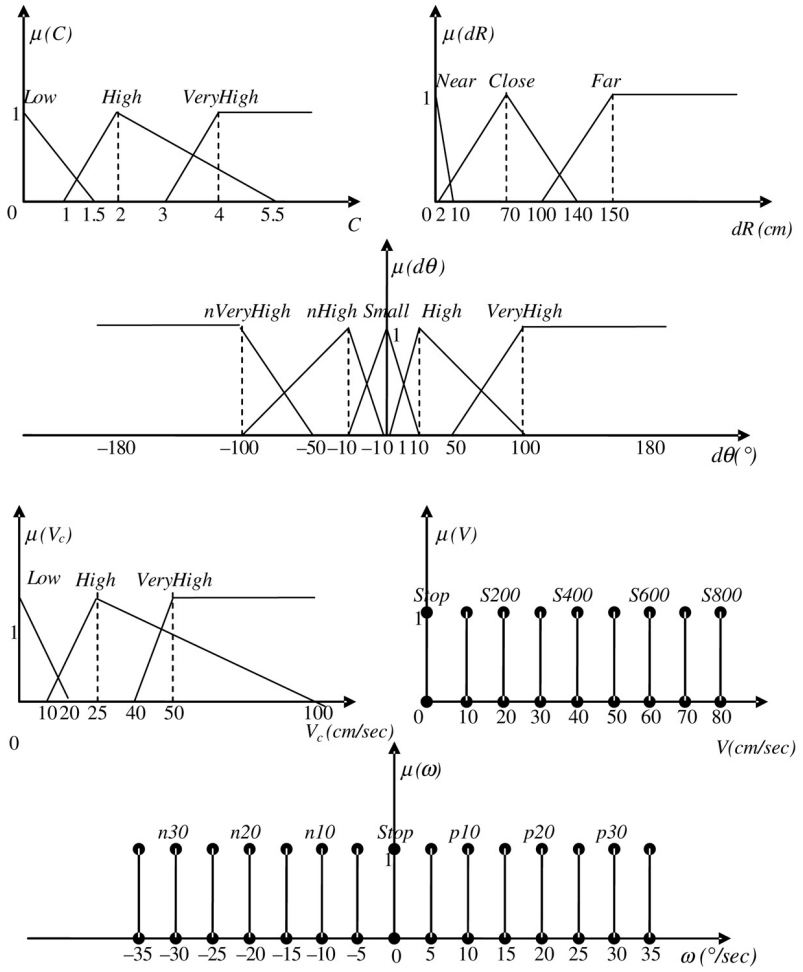

Fig. 7. Membership functions of path tracker parameters.

smooth and almost continuous manner. When the curvature is sharp, the controller decreases the speed and outputs the needed rotational speed in the right direction to make the turn smoothly. When the curvature is smooth, the robot will slow up and the rotational speed is small so as to stay on track. For example the rule

IF C is *High* THEN V is $S300$

sets the speed to 300 mm/s when the curvature value is high no matter what the other values at the inputs are if it is the only inference rule that is activated by the inputs. Otherwise, the COGS defuzzification method mentioned above is used again to calculate the outputs. The value of rotational speed is not affected by this rule. Another example is when the current velocity V_c is high and either $\mathrm{d}\theta$ or C is high, the robot should slow down first before making the turn. Figs. 8 and 9 show the input–output characteristics at C and V_c both set to zero. Note the symmetry with respect to the plane that satisfies $\mathrm{d}\theta = 0$.

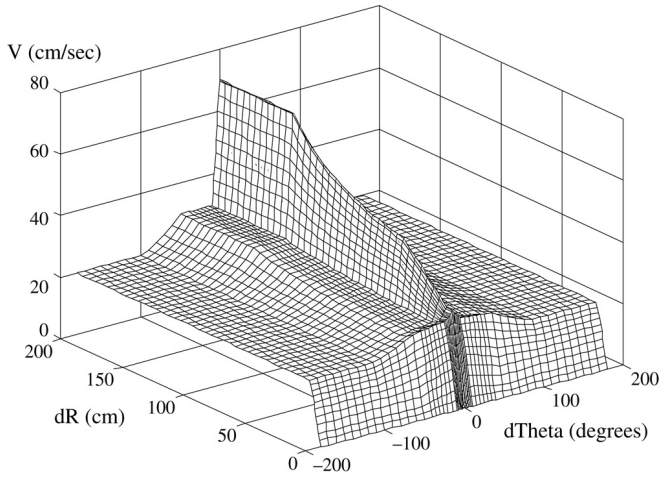

Fig. 8. V output when C and V_c are zero.

5. Classical Lyapunov derived control

In this section, a more theoretical approach based on a Lyapunov function is briefly described. It was implemented in real time on the P3-AT to compare its performance to the fuzzy controller. The approach used is based on an error model of the kinematical model [18].

Knowing that the kinematical model of a differentially steered wheeled mobile robot in Cartesian coordinates is given by Eq. (1), the objective is to track a virtual reference robot:

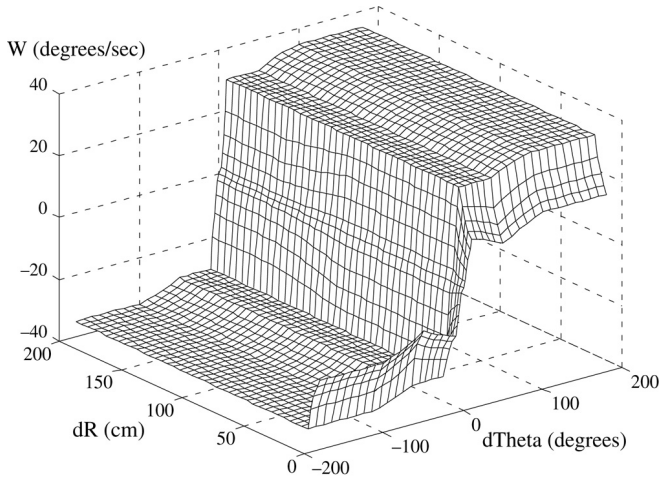

Fig. 9. Angular speed output when C and V_c are zero.

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix}. \tag{5}$$

Then, three error variables e_x, e_y, and e_θ that correspond to the instantaneous errors in posture variables are chosen as:

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}. \tag{6}$$

These errors are the errors in posture with respect to the local frame of reference of the robot (Fig. 10). The transformation matrix converts global coordinates to local coordinates. Calculation of the derivatives of the errors using the constraint $\dot{x}_r \sin\theta_r = \dot{y}_r \cos\theta_r$ and with $\theta_e = e_\theta = \theta_r - \theta$, we obtain:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} v + \begin{bmatrix} e_y \\ -e_x \\ -1 \end{bmatrix} \omega + \begin{bmatrix} v_r \cos e_\theta \\ v_r \sin e_\theta \\ \omega_r \end{bmatrix}. \tag{7}$$

From Eq. (7) above, the aim of a control law is to make the errors converge to zero. The proposed velocity inputs v_f and ω_f of the control law [18] are:

$$\begin{aligned} v_f &= v_r \cos e_\theta + K_x e_x \\ \omega_f &= \omega_r + V_r K_y e_y + K_\theta \sin e_\theta. \end{aligned} \tag{8}$$

By substituting v_f and ω_f in the errors of Eq. (7), we get:

$$\begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} e_y(\omega_r + v_r(K_y e_y + K_\theta \sin e_\theta)) - K_x e_x \\ -e_x(\omega_r + v_r(K_y e_y + K_\theta \sin e_\theta)) + v_r \sin e_\theta \\ -v_r(K_y e_y + K_\theta \sin e_\theta) \end{bmatrix}. \tag{9}$$

If the Lyapunov energy function V_0 is chosen to be:

$$V_0 = \frac{1}{2}(e_x^2 + e_y^2) + \frac{1 - \cos e_\theta}{K_y} \tag{10}$$

by deriving V_0 with respect to time, we get:

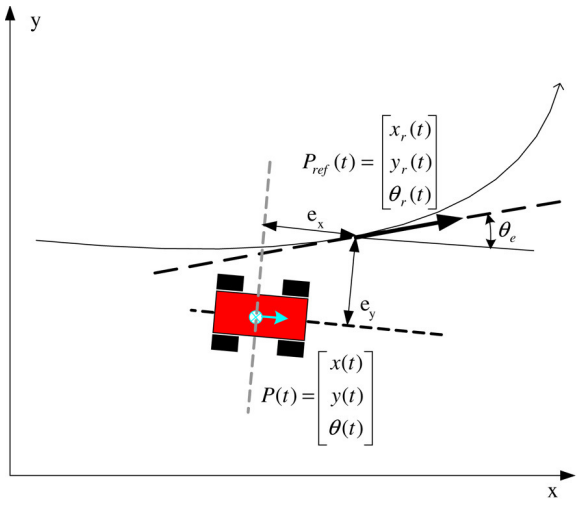

Fig. 10. Parameters of the error model.

Fig. 11. The Pioneer 3-AT mobile robot in the lab.

$$\begin{aligned} \dot{V}_0 &= e_x\dot{e}_x + e_y\dot{e}_y + [-(\omega_r + v_r(K_y y_e \\ &\quad + K_\theta \sin\theta_e))x_e + v_r \sin\theta_e]e_y \\ &\quad + [-v_r(K_y y_e + K_\theta \sin\theta_e)] \sin\theta_e / K_y \\ &= -K_x e_x^2 - \frac{K_\theta \sin^2 e_\theta}{K_y} \le 0. \end{aligned} \tag{11}$$

Given that K_x, K_y, and K_θ are all positive constants, the above inequality would be satisfied and the system with the control law would be stable.

6. Real time implementation

The path tracking FLC shown above is implemented using the C++ *Free Fuzzy Logic Library* (FFLL) along with the *Activmedia Robotic Interface for Application* (ARIA) library that provides extensive methods to control the Pioneer robot, communicate with it, and obtain its sensor information. The fuzzy controllers are specified in Fuzzy Control Language (FCL) files using the IEC 61131-7 industrial standard [16]. The FFLL contains methods to read FCL files that contain the

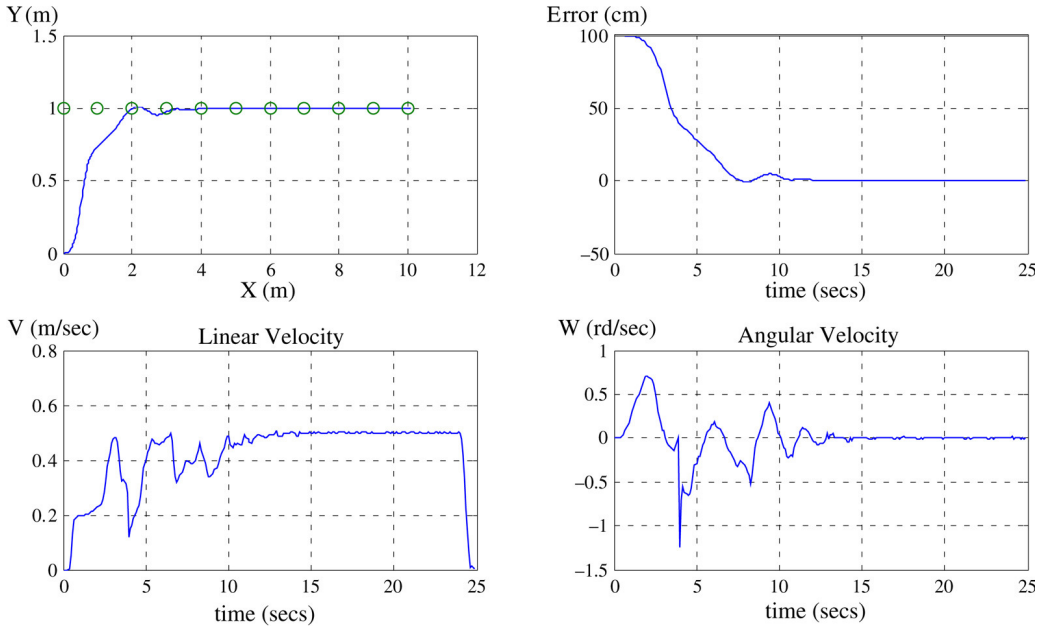

Fig. 12. Fuzzy controller response to a unit step trajectory in real time.

input and output information and membership functions, the defuzzification method(s), and the inference rules. The FFLL also contains the methods needed to calculate the fuzzy outputs once the FCL files are read. The real time implementation was tested on the Pioneer 3-AT (Fig. 11) four-wheel differentially driven mobile robot used for all terrain navigation. Through its serial port, it can be controlled either by an onboard computer or by a computer through a wireless serial or TCP/IP connection. All the information from the onboard sensors such as the encoders, sonars, and gyroscope can be obtained through this connection and the reference linear and rotational speeds are sent through ARIA. The lower level controllers of the motors are separate with a PID controller for each motor alone. Only the values of the gains of the PID controllers can be modified to cope with the weight of the robot, but only reference speeds or positions can be controlled.

7. Simulation and experimental results

The path tracking controller has been tested on both the SRI simulator with the parameters of the P3-AT robot as well as on the real robot. The controller corrects the path of the robot with respect to the position obtained by the encoders. Therefore some error due to localization will be unavoidable as the robot departs from its initial position. The tests in real time were conducted in an indoor environment, with the robot equipped with the wheels used for navigation on rough terrain due to the unavailability of wheels used to navigate on smooth surfaces. This renders the motion of the robot sluggish, and would degrade the performance in terms of precision. To remedy this problem, the maximum speed was limited to 0.5 m/s. Tests on the same sequences of waypoints were conducted using the classical controller described in Section 5. The reference trajectory for the classical controller was obtained by interpolating the waypoints. The best combination of gains of K_x, K_y, and K_θ in Eq. (8) for real time performance are 2.5, 0.75, and 1.41. Tests have been conducted on several sequences of waypoints to validate the performance of the fuzzy controller. In all test cases, the robot is initially oriented with the x-axis in the positive sense at zero heading. The first tests were done on a sequence of waypoints that lie on step functions of different magnitudes in the x–y plane to assess the response to lateral discontinuities. To assess the response to directional discontinuities, the sequences of waypoints are chosen to lie on straight lines heading at 45°, 90°, 135°, and 180°. The performance of the controller on a sequence of waypoints that lie on a sine wave is also tested. For the same sine wave, the variation of the distance between waypoints demonstrates the variation in precision. The error in position at any instant of time is the distance from the position of the robot at the defined instant to the nearest point on the reference trajectory. Since no continuous trajectory is available, the error is calculated with respect to the straight line whenever the waypoints lie on a straight line. If the waypoints lie on a sine wave trajectory, the error is calculated with respect to the sine wave trajectory, since the ideal interpolation of the waypoints would approximate a sine wave. In all the figures, the dashed lines represent the virtual reference trajectory, and the waypoints are the small circles.

The response of a controller to a lateral discontinuity as in the case of a step trajectory can be evaluated by the rise distance and steady state error. The rise distance in this paper is the distance traveled by the robot in the x direction to reach 70% of the step size (magnitude of the step function). The steady state error is the maximum error in y with respect to the reference after the robot has joined the trajectory. Fig. 12 shows the performance of the fuzzy controller on a unit step reference trajectory. The rise distance is 0.69 m and the steady state error is less than 0.35 cm. The error starts at 100 cm and attenuates

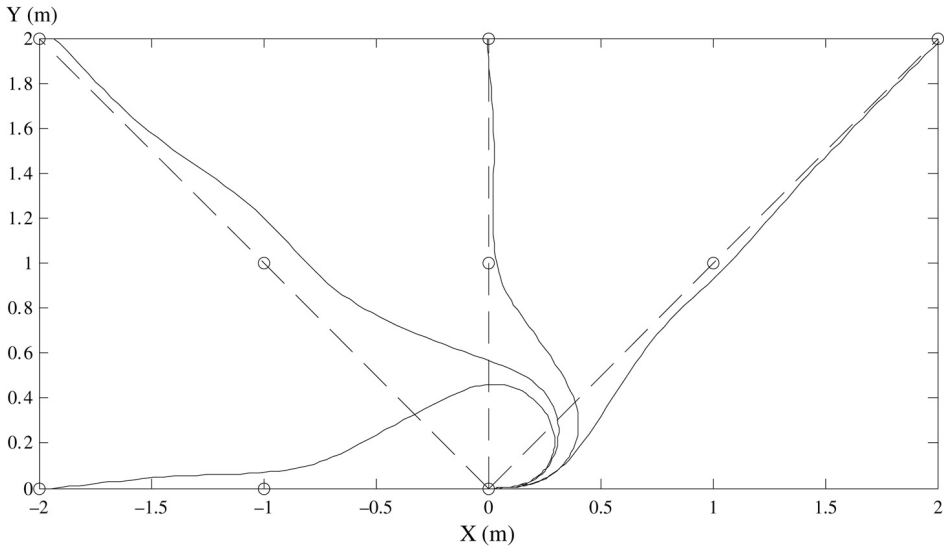

Fig. 13. Fuzzy controller responses to directional discontinuities in real time.

Table 2
Step responses of the fuzzy controller

Step size (m)	Simulation		Real time implementation	
	Rise distance (m)	Steady state error (cm)	Rise distance (m)	Steady state error (cm)
0.10	0.69	0.35	0.94	0.28
0.20	0.73	0.36	0.92	0.32
0.30	0.77	0.48	0.81	0.32
0.50	0.75	0.25	0.89	0.33
0.75	0.75	0.33	0.86	0.36
1.00	0.88	0.35	0.98	0.40

Table 3
Step responses of the classical controller

Step size (m)	Rise distance (m)	Steady state error (cm)
0.10	2.25	1.65
0.20	2.32	1.58
0.30	2.21	1.80
0.50	2.25	1.60
0.75	2.14	1.76
1.00	2.15	1.70

Table 4
Fuzzy controller responses to directional discontinuities

Angular discontinuity (°)	Simulation		Real time implementation	
	Error overshoot (cm)	Convergence distance (m)	Error overshoot (cm)	Convergence distance (m)
45	9	0	16.7	0.66
90	40.1	0.84	26	1.13
135	42.0	0.76	40	1.45
180	45.8	0.80	46	1.31

to the steady state error. Note that the linear velocity is at its maximum of 0.5 m/s and the rotational velocity oscillates very smoothly around zero.

In Table 2, the values of rise distance and steady state error are shown for different magnitudes of step reference trajectories in real time and in simulation. It is noted from the results of Table 2 that the rise distance and the steady state error are independent of the magnitude of the lateral discontinuity. The rise distance and steady state errors can be reduced by decreasing the distance between the waypoints, as discussed later on in this section. The simulation and real time responses are close to each other. From Table 3, we can deduce that the responses to a lateral discontinuity obtained by the classical controller of Section 5 with the gains specified above are significantly inferior to those of the fuzzy controller. This is due to the fact that the classical controller regulates five parameters: the errors in x and y, the errors in linear and angular velocities, and the error in orientation. This renders the fuzzy controller way more responsive to lateral discontinuities.

The response to directional discontinuities can be evaluated by the error overshoot and the convergence distance. The error overshoot is the maximum deviation from the reference straight line that connects the waypoints. The convergence distance is the distance the robot travels along the straight line before it becomes less than 10 cm. The responses of the fuzzy controller directional discontinuities at 45°, 90°, 135°, and 180° are shown in Fig. 13.

The performance of the responses to directional discontinuities is shown by the numerical results of Table 4. The performance in simulation is better than in real time with regards to convergence distance but is slightly inferior in error overshoot. The response can be improved at the expense of speed by taking waypoints at smaller distances from each other. The performance results of the classical controller in real time (Table 5) are inferior to those of the fuzzy controller for the same considerations of the case of lateral discontinuities discussed above. At a 180 degrees directional discontinuity, the classical controller drives the robot backwards till

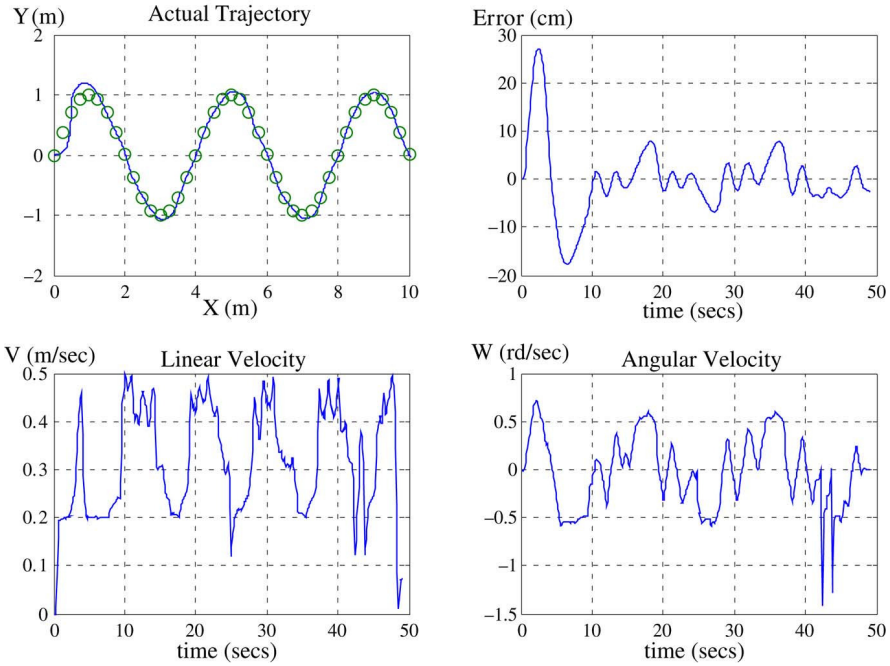

Fig. 14. Fuzzy controller execution of a sine wave trajectory in real time.

Table 5
Classical controller responses to directional discontinuities

Angular discontinuity (°)	Error overshoot (cm)	Convergence distance (m)
45	33	2.34
90	42	3.93
135	41	0.38
180	47	>5

Table 6
Effect of waypoint inter-distance on precision

Distance in x (m)	Error overshoot (cm)	Steady state error (cm)
0.2500	15.2	6.40
0.2000	15.0	5.76
0.1250	16.5	4.65
0.1000	16.9	3.62
0.0625	13.3	2.10

before the last waypoint where it turns and drives forward till the destination position, and the overshoot occurs towards the end of the trajectory. This again demonstrates the robustness of the fuzzy controller to abrupt discontinuities in reference trajectories.

If the waypoints lie on a sine wave trajectory (Fig. 14), the best interpolation of the waypoints would be a sine wave trajectory. Consequently, the error in steady state is calculated with respect to a virtual sine wave trajectory. This would be a good assessment of the fuzzy controller in terms of smoothness in the trajectory traced for a certain sequence of waypoints. The sine wave of Fig. 14 has an amplitude of 1 m and a period of 4 m. The waypoints are 25 cm apart in the x direction and lie on the sine wave. The 28 cm of error overshoot at the beginning is due to the directional discontinuity, since the robot is initially at zero heading. The maximum error in steady state is inferior to 8 cm. At the peaks, the linear velocity reaches a minimum and the angular velocity attains its maximum. The inverse happens at zero crossings. This largely resembles the behavior of a human driver on a similar trajectory. The fuzzy inference rules of the fuzzy controller reflect this behavior. In the case of a classical controller (Fig. 15), the error overshoot due to the directional discontinuity is around 27 cm, and the maximum error in steady state is 11 cm. The velocities of the fuzzy controller are smoother and the overall trajectory is smoother.

As has been mentioned before, the precision of the fuzzy controller can be ameliorated by decreasing the distance between consecutive waypoints. Tests were conducted on the same sine wave virtual reference trajectory discussed above, but with the distance in x between the waypoints varied from 0.25 to 0.0625 m. The results in Table 6 were obtained by tests on the simulator. Change in overshoot is not very significant, but the decrease in the maximum value of the steady state error is clear as the distance is decreased. The size of the robot is 50 cm × 50 cm. A precision of 2.1 cm can be obtained, at the expense of traveling at lower speeds nevertheless.

8. Conclusion

The behavioral fuzzy logic path tracking controller that has been implemented proved to be highly reliable and robust in terms of precision and speed of response. The reference path needs only be a set of waypoints and there is no need for calculating a continuous reference trajectory. It has some interesting advantages over other approaches

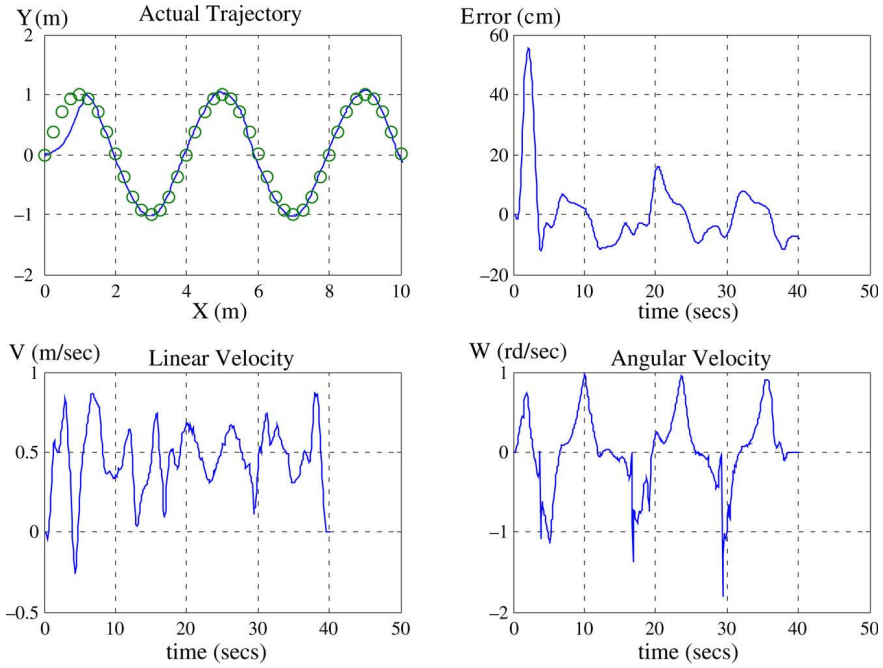

Fig. 15. Classical controller execution of a sine wave trajectory in real time.

for trajectory following. The fuzzy controller controls robot motion along the discrete waypoints of a path that can be calculated by some higher level optimization algorithm. Position control is performed successively along the waypoints till the desired position that corresponds to the last waypoint on the path is reached. Despite the fact that fuzzy logic control is not based on a precise mathematical model, it is robust and flexible. A lower level controller can be implemented independent from the path tracking problem or other behaviors that will be integrated. Future work will include a path tracking controller for a trajectory with timed parameters, a stability analysis, and integrating path tracking with dynamic obstacle avoidance behavior. Other future work includes the modification of the controller for rugged terrain navigation.

References

[1] J.R. Zhang, S.J. Xu, A. Rachi, Path tracking control of vehicles based on Lyapunov approach, in: Proceedings of the American Control Conference, vol. 3, Anchorage, AK, USA, 2002, pp. 2132–2137.

[2] R.M. DeSantis, R. Hurteau, O. Alboui, B. Lesot, Experimental stabilization of tractor and tractor-trailer like vehicles, in: Proceedings of the IEEE International Symposium on Intelligent Control, Vancouver, Canada, 2002, pp. 188–193.

[3] K.C. Koh, H.S. Cho, Wheel servo control based on feedforward compensation for an autonomous mobile robot, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems'95, in: Human Robot Interaction and Cooperative Robots, vol. 3, Pittsburgh, PA, USA, 1995, pp. 454–459.

[4] L. Caracciolo, A. de Luca, S. Iannitti, Trajectory tracking control of a four-wheel differentially driven mobile robot, in: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 4, Detroit, MI, USA, 1999, pp. 2632–2638.

[5] J.R. Zhang, S.J. Xu, A. Rachid, Sliding mode controller for automatic steering of vehicles, in: The 27th Annual Conference of the IEEE Industrial Electronics Society'01, vol. 3, Denver, CO, USA, 2001, pp. 2149–2153.

[6] J.R. Zhang, S.J. Xu, A. Rachid, Sliding mode controller for automatic path tracking of vehicles, in: Proceedings of the IEEE American Control Conference, vol. 5, Anchorage, AK, USA, 2002, pp. 3974–3979.

[7] A. Shaout, M.A. Jarrah, H. Al-Araji, K. Al-Tell, A nonlinear optimal four wheels steering controller, in: Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, vol. 3, Lansing, MI, USA, 2000, pp. 1426–1429.

[8] S.X. Yang, H. Li, M. Meng, Fuzzy control of a behavior-based mobile robot, in: The 12th IEEE International Conference on Fuzzy Systems'03, vol. 1, 2003, pp. 319–324.

[9] H. Xu, S.X. Yang, Tracking control of a mobile robot with kinematic and dynamic constraints, in: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation'01, Alberta, Canada, 2001, pp. 125–130.

[10] W. Weiguo, C. Huitang, W. Yuejuan, Backstepping design for path tracking of mobile robots, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems'99, vol. 3, Kyongju, Korea, 1999, pp. 1822–1827.

[11] A. Ollero, G. Heredia, Stability analysis of mobile robot path tracking, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems'95, Pittsburgh, PA, USA, in: Human Robot Interaction and Cooperative Robots, vol. 3, 1995, pp. 461–466.

[12] K. Yoshizawa, H. Hashimoto, M. Wada, S. Mori, Path tracking control of mobile robots using a quadratic curve, in: Proceedings of the IEEE Intelligent Vehicles Symposium, Tokyo, Japan, 1996, pp. 58–63.

[13] M. Davidson, V. Bahl, The scalar ε-controller: A spatial path tracking approach for ODV, Ackerman, and differentially-steered autonomous wheeled mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, Seoul, Korea, 2001, pp. 175–180.

[14] X. Yang, H. Kezhong, M. Guo, B. Zhang, An intelligent predictive control approach to path tracking problem of autonomous mobile robot, in: 1998 IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, San Diego, CA, USA, 1998, pp. 3301–3306.

[15] D. Driankov, A. Saffiotti, Mobile robot path tracking and visual target tracking using fuzzy logic, in: Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, New York, 2001.

[16] International Technical Commission (IEC), Devices, IEC1131 - Programmable Controllers, Fuzzy Control Programming, Part 7, http://www.fuzzytech.com/binaries/ieccd1.pdf, 1997 (last consulted: 20 July 2005).
[17] S.S.F. Farinwata, P. Dimitar, R. Langari, Fuzzy Control: Synthesis and Analysis, John Wiley and Sons, New York, 2000.
[18] Y. Kanayama, Y. Kimura, F. Miyazaki, T. Noguchi, A stable tracking control method for an autonomous mobile robot, in: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, Cincinnati, OH, USA, pp. 384–389.

Elie Maalouf is a master's student at Ecole de Technologie Superieure in Montreal, Canada. He received a bachelor's degree of engineering in electrical engineering at the American University of Beirut, Lebanon. His research interests include mobile robot navigation and control and optimization algorithms.

Maarouf Saad received a bachelor and a master degrees in electrical engineering from Ecole Polytechnique of Montreal respectively in 1982 and 1984. In 1988, he received a Ph.D. from McGill University in electrical engineering. He joined Ecole de technologie superieure in 1987 where he is teaching control theory and robotics courses. His research is mainly in nonlinear control, and optimization applied to robotics and flight control system.

Hamadou Saliah-Hassane has an Electronics degree from Cheik Anta Diop University (Senegal, 1976), a Bachelor's degree in Engineering, and a Master of Applied Science degree from École Polytechnique de Montréal and a Ph.D. degree in Computer Aided Analysis and Design from the Electrical and Computer Engineering at McGill University in Montreal. He is currently teaching Informatics and Computer Networks at Télé-université in Montreal. His research interests are on Virtual and Remote Laboratories including Grid and Web services and Distributed Systems and High Performance Computing for Electrical Engineering.